CLAIMS

1. One or more electronically-accessible media comprising electronically-executable instructions that comprise at least part of an operating system, the at least part of the operating system including:

an application programming interface that is capable of creating callback-type dynamic function tables, each callback-type dynamic function table including a begin address, an end address, and a callback function; wherein the operating system calls the callback function when an exception occurs within an address range bounded by the begin address and the end address, the callback function requesting exception handling and/or unwind information from a runtime component.

2. The one or more electronically-accessible media as recited in claim 1, wherein the callback function returns from the runtime component data for code address-to-pointer information.

3. The one or more electronically-accessible media as recited in claim 2, wherein the data for code address-to-pointer information comprises a reference to the code address-to-pointer information.

4. The one or more electronically-accessible media as recited in claim 2, wherein the data for code address-to-pointer information comprises the code address-to-pointer information.

5. The one or more electronically-accessible media as recited in claim 4, wherein the code address-to-pointer information comprises a start address, a final address, and an unwind pointer.

6. One or more electronically-accessible media comprising electronically-executable instructions that utilize an application programming interface, the application programming interface facilitating creation of callback-type dynamic function tables; each callback-type dynamic function table including a begin address, an end address, and a callback function, each callback-type dynamic function table corresponding to a code heap that stores code for a plurality of functions in a runtime environment; wherein interaction between the runtime environment and an operating system is precipitated upon calling the callback function to acquire exception handling and/or unwind information.

7. The one or more electronically-accessible media as recited in claim 6, wherein the interaction between the runtime environment and the operating system comprises (i) the operating system requesting exception handling and/or unwind information from the runtime environment and (ii) the runtime environment providing the exception handling and/or unwind information to the operating system.

8. The one or more electronically-accessible media as recited in claim 6, wherein the begin address and the end address of each callback-type dynamic function table comprises a begin address and an end address of a respective code heap to which a respective callback-type dynamic function table corresponds.

9. The one or more electronically-accessible media as recited in claim 6, wherein the exception handling and/or unwind information comprises data for code address-to-pointer information for a particular function of the plurality of functions.

10. The one or more electronically-accessible media as recited in claim 9, wherein the particular function of the plurality of functions is ascertained using an instruction pointer that is passed as an argument of the callback function.

11. The one or more electronically-accessible media as recited in claim 6, wherein each callback-type dynamic function table is created, at least in part, by adding each callback-type dynamic function table to a linked list of dynamic function tables.

12. The one or more electronically-accessible media as recited in claim 11, wherein the linked list of dynamic function tables includes one or more non-callback-type dynamic function tables.

13. One or more electronically-accessible media comprising electronically-executable instructions that comprise at least part of a runtime component, the runtime component adapted to manage exception handling and/or unwind information for a plurality of functions that are executing as part of a runtime environment, the runtime component capable of providing data for code address-to-pointer information to an operating system when an exception occurs with respect to a particular function of the plurality of functions.

14. The one or more electronically-accessible media as recited in claim 13, wherein the data for code address-to-pointer information comprises a reference to the code address-to-pointer information for the particular function of the plurality of functions.

15. The one or more electronically-accessible media as recited in claim 13, wherein the data for code address-to-pointer information comprises the code address-to-pointer information for the particular function of the plurality of functions.

16. The one or more electronically-accessible media as recited in claim 15, wherein the code address-to-pointer information comprises a start address, a final address, and an unwind pointer for the particular function of the plurality of functions.

17. An electronic device comprising:

a runtime environment that is managing code for a plurality of functions; and

an operating system that is managing a linked list of dynamic function tables that are searched when an exception occurs, the operating system adapted to call a callback function as indicated by a dynamic function table of the linked list of dynamic function tables to request that the runtime environment provide exception handling and/or unwind information for at least one function of the plurality of functions;

wherein the runtime environment is capable of providing to the operating system the exception handling and/or unwind information for the at least one function of the plurality of functions responsive to the callback function.

18. The electronic device as recited in claim 17, wherein the dynamic function table comprises a callback dynamic function table.

19. The electronic device as recited in claim 18, wherein the linked list of dynamic function tables comprises at least one sorted dynamic function table and at least one unsorted dynamic function table.

20. The electronic device as recited in claim 17, wherein the dynamic function table corresponds to more than one function of the plurality of functions.

21. The electronic device as recited in claim 17, wherein the dynamic function table includes a begin address entry and an end address entry that reflect a begin address and an end address, respectively, of a code heap that includes more than one function of the plurality of functions, including the at least one function of the plurality of functions.

22. The electronic device as recited in claim 17, wherein the runtime environment comprises at least one just-in-time (JIT) compiler.

23. The electronic device as recited in claim 17, wherein the exception handling and/or unwind information comprises data for code address-to-pointer information for the at least one function of the plurality of functions.

24. The electronic device as recited in claim 23, wherein the data for code address-to-pointer information for the at least one function of the plurality of functions directly or indirectly provides to the operating system a start address, a final address, and an unwind pointer for the at least one function of the plurality of functions.

25. One or more electronically-accessible media comprising a data structure, the data structure comprising:

a begin address;

an end address; and

a callback function that, when called, returns from a runtime environment exception handling and/or unwind information for a function associated with at least one address that is between the begin address and the end address.

26. The one or more electronically-accessible media as recited in claim 25, wherein the begin address and the end address reflect a begin address and an end address, respectively, for a code heap of the runtime environment.

27. The one or more electronically-accessible media as recited in claim 26, wherein the code heap includes code for a plurality of functions.

28. The one or more electronically-accessible media as recited in claim 27, wherein the callback function, when called, is capable of returning from the runtime environment exception handling and/or unwind information for any function of the plurality of functions.

29. The one or more electronically-accessible media as recited in claim 25, wherein the callback function is called by an operating system to enable the operating system to unwind a stack.

30. The one or more electronically-accessible media as recited in claim 25, wherein the callback function accepts as input the at least one address that is associated with the function, the at least one address comprising an instruction pointer.

31. The one or more electronically-accessible media as recited in claim 25, wherein the callback function accepts as input a reference to at least part of the data structure.

32. The one or more electronically-accessible media as recited in claim 25, wherein the exception handling and/or unwind information comprises data for code address-to-pointer information for the function that is associated with the at least one address.

33. The one or more electronically-accessible media as recited in claim 25, wherein the begin address, the end address, and the callback function together comprise a callback dynamic function table.

34. The one or more electronically-accessible media as recited in claim 33, wherein the data structure further comprises:

a plurality of dynamic function tables, the plurality of dynamic function tables including the callback dynamic function table.

35. An electronic device comprising:

at least one processor; and

one or more media in operative communication with the at least one processor, the one or more media including a data structure comprising:

a begin address value;

an end address value; and

a callback function that, when called, returns from a runtime environment exception handling and/or unwind information for a function associated with at least one address that is between the begin address value and the end address value.

36. The electronic device as recited in claim 35, wherein the one or more media further include a code heap managed by the runtime environment, the code heap having a begin address and an end address; the begin address value and the end address value reflecting the begin address and the end address, respectively, of the code heap.

37. The electronic device as recited in claim 36, wherein the code heap includes code for a plurality of functions.

38. The electronic device as recited in claim 37, wherein the callback function, when called, is capable of returning from the runtime environment exception handling and/or unwind information for any function of the plurality of functions responsive to the at least one address.

39. The electronic device as recited in claim 35, wherein the one or more media further include a stack; and wherein the callback function is called by an operating system to enable the operating system to unwind the stack.

40. One or more electronically-accessible media comprising electronically-executable instructions that include:

a callback function, the callback function accepting as input an instruction pointer that is associated with an address of a function from a runtime environment and producing as output data for code address-to-pointer information for the function having the address that is associated with the instruction pointer;

wherein the callback function may be called by an operating system and implemented by the runtime environment.

41. The one or more electronically-accessible media as recited in claim 40, wherein the callback function further accepts as input a reference to a callback dynamic function table for context.

42. The one or more electronically-accessible media as recited in claim 41, wherein the callback dynamic function table includes a begin address and an end address that define an address range that includes the instruction pointer.

43. The one or more electronically-accessible media as recited in claim 40, wherein the data for code address-to-pointer information that is output by the callback function is provided from the runtime environment to the operating system.

44. The one or more electronically-accessible media as recited in claim 40, wherein the data for code address-to-pointer information comprises a reference to the code address-to-pointer information.

45. The one or more electronically-accessible media as recited in claim 44, wherein the referenced code address-to-pointer information comprises a start address, a final address, and an unwind pointer for the function having the address that is associated with the instruction pointer.

46. The one or more electronically-accessible media as recited in claim 40, wherein the data for code address-to-pointer information comprises the code address-to-pointer information.

47. The one or more electronically-accessible media as recited in claim 40, wherein the callback function is adapted to be called by the operating system during an exception handling procedure.

48. The one or more electronically-accessible media as recited in claim 40, wherein the callback function is implemented, at least partially, by the runtime environment by inspecting a code header for the function having the address that is associated with the instruction pointer.

49. The one or more electronically-accessible media as recited in claim 48, wherein the data for code address-to-pointer information is derived using the code header.

50. The one or more electronically-accessible media as recited in claim 48, wherein the code header is ascertained using a heap structure contents of a code heap that includes code for the function having the address that is associated with the instruction pointer.

51. One or more electronically-accessible media comprising at least part of an operating system that is configured to request from a runtime environment exception handling and/or unwinding information for functions that are managed by the runtime environment.

52. The one or more electronically-accessible media as recited in claim 51, wherein the at least part of the operating system is further configured to request from the runtime environment the exception handling and/or unwinding information after an exception is discovered.

53. The one or more electronically-accessible media as recited in claim 51, wherein the at least part of the operating system is further configured to request the exception handling and/or unwinding information from the runtime environment responsive to locating a dynamic function table having a callback function.

54. The one or more electronically-accessible media as recited in claim 53, wherein the dynamic function table having the callback function corresponds to at least two functions compiled in the runtime environment by a just-in-time (JIT) compiler.

55. An electronic device comprising:

at least one processor; and

one or more media including processor-executable instructions that are capable of being executed by the at least one processor, the processor-executable instructions adapted to cause the electronic device to perform actions comprising:

initializing a code heap having a begin address and an end address for a runtime; and

creating a dynamic function table corresponding to the code heap, the dynamic function table including the begin address, the end address, and a callback function.

56. The electronic device as recited in claim 55, wherein the initializing action comprises:

initializing the code heap by at least one of the runtime and an operating system.

57. The electronic device as recited in claim 55, wherein the processor-executable instructions are adapted to cause the electronic device to perform a further action comprising:

installing the dynamic function table as part of a linked list of dynamic function tables.

58. The electronic device as recited in claim 57, wherein the processor-executable instructions are adapted to cause the electronic device to perform a further action comprising:

removing the dynamic function table from the linked list of dynamic function tables.

59. The electronic device as recited in claim 55, wherein the processor-executable instructions are adapted to cause the electronic device to perform further actions comprising:

initiating the callback function by an operating system responsive to an exception; and

providing data for code address-to-pointer information from the runtime to the operating system.

60. The electronic device as recited in claim 55, wherein the processor-executable instructions are adapted to cause the electronic device to perform a further action comprising:

storing, by the runtime, code for a plurality of functions in the code heap.

61. One or more electronically-accessible media comprising electronically-executable instructions that, when executed, direct an electronic device to perform actions comprising:

initializing a code heap for a runtime, the code heap having a begin address and an end address; and

creating a dynamic function table corresponding to the code heap, the dynamic function table including the begin address, the end address, and a callback function.

62. One or more electronically-accessible media comprising electronically-executable instructions that, when executed, direct an electronic device to perform actions comprising:

receiving at a runtime an instruction pointer with a request for data for code address-to-pointer information;

ascertaining a runtime function associated with the instruction pointer;

accessing a code header of the ascertained runtime function to extract data for code address-to-pointer information for the ascertained runtime function; and

providing the extracted data for code address-to-pointer information from the runtime.

63. The one or more electronically-accessible media comprising the electronically-executable instructions that, when executed, direct an electronic device to perform the actions as recited in claim 62, wherein the action of providing comprises the action of:

providing from the runtime a pointer to the code address-to-pointer information for the ascertained runtime function.

64. The one or more electronically-accessible media comprising the electronically-executable instructions that, when executed, direct an electronic device to perform the actions as recited in claim 62, wherein the action of providing comprises the action of:

providing from the runtime to an operating system the extracted data for code address-to-pointer information.

65. An electronic device comprising:

at least one processor; and

one or more media including processor-executable instructions that are capable of being executed by the at least one processor, the processor-executable instructions adapted to cause the electronic device to perform actions comprising:

locating a callback dynamic function table having a callback function;

initiating, by an operating system, the callback function to interact with a runtime;

receiving data for code address-to-pointer information from the runtime responsive to the initiating; and

using the received data for code address-to-pointer information to attain information for at least one of exception handling and stack unwinding.

66. The electronic device as recited in claim 65, wherein:

the receiving action comprises:

receiving a reference to code address-to-pointer information from the runtime; and

the using action comprises:

using the reference to attain code address-to-pointer information, including an unwind pointer; and

using the unwind pointer to attain unwind information.

67. The electronic device as recited in claim 65, wherein the processor-executable instructions are adapted to cause the electronic device to perform a further action comprising:

discovering an exception from a runtime function.

68. The electronic device as recited in claim 65, wherein the processor-executable instructions are adapted to cause the electronic device to perform a further action comprising:

searching a dynamic function table linked list using an instruction pointer;

wherein the locating action comprises:

locating the callback dynamic function table having the callback function from the dynamic function table linked list using the instruction pointer, the callback dynamic function table including a begin address that is less than and an end address that is greater than the instruction pointer.

69. The electronic device as recited in claim 68, wherein the processor-executable instructions are adapted to cause the electronic device to perform further actions comprising:

receiving at the runtime, responsive to the initiating, the instruction pointer with a request for the data for code address-to-pointer information;

ascertaining a runtime function associated with the instruction pointer;

accessing a code header of the ascertained runtime function to extract the data for code address-to-pointer information for the ascertained runtime function; and

providing, from the runtime to the operating system, the data for code address-to-pointer information.

70. One or more electronically-accessible media comprising electronically-executable instructions that, when executed, direct an electronic device to perform actions comprising:

locating a callback dynamic function table having a callback function;

initiating, by an operating system, the callback function to interact with a runtime;

receiving, at the operating system, data for code address-to-pointer information from the runtime responsive to the initiating; and

using the received data for code address-to-pointer information to attain information for at least one of exception handling and stack unwinding.

71. An arrangement comprising:

means for initializing a code heap for a runtime, the code heap having a begin address and an end address; and

means for creating a dynamic function table corresponding to the code heap, the dynamic function table including the begin address, the end address, and a callback function.

72. An arrangement comprising:

means for locating a dynamic function table having a callback function;

means for initiating the callback function to interact with a runtime environment;

means for receiving data for code address-to-pointer information from the runtime environment responsive to the callback function initiation; and

means for using the received data for code address-to-pointer information to attain information for at least one of exception handling and stack unwinding.